AUSTEN
TECH

MSE REPORT

Submitted by: Dhruv Kumar Sharma

Date: 11-March-2025

1. Introduction

Predicting student performance is crucial in educational analytics. This project utilizes machine learning techniques to analyze student data and predict performance based on study hours, previous scores, and other academic factors. The Random Forest Classifier is used to train a predictive model, and its accuracy is evaluated to determine its effectiveness.

Grade 12

This project explores data preprocessing, feature selection, and model evaluation techniques, making it an essential application of data science in education.

METHODOLOGY-

a

Step 1: Data Collection & Preprocessing

- The dataset is loaded from a CSV file containing StudentID, StudyHours, PreviousScores, and FinalExamScore.
- Any missing values are removed.
- Categorical variables (if present) are encoded using LabelEncoder.
- Features are standardized using StandardScaler to improve model performance.

Step 2: Splitting the Data

- The dataset is split into training (80%) and testing (20%) sets using train_test_split().

Step 3: Model Training

- A Random Forest Classifier with 100 estimators is trained on the processed data.

Step 4: Model Evaluation

- Predictions are made on the test set.
- The model's accuracy is computed using accuracy_score().
- A classification report is generated to evaluate the model’s precision, recall, and F1-score.

Step 5: Feature Importance Analysis

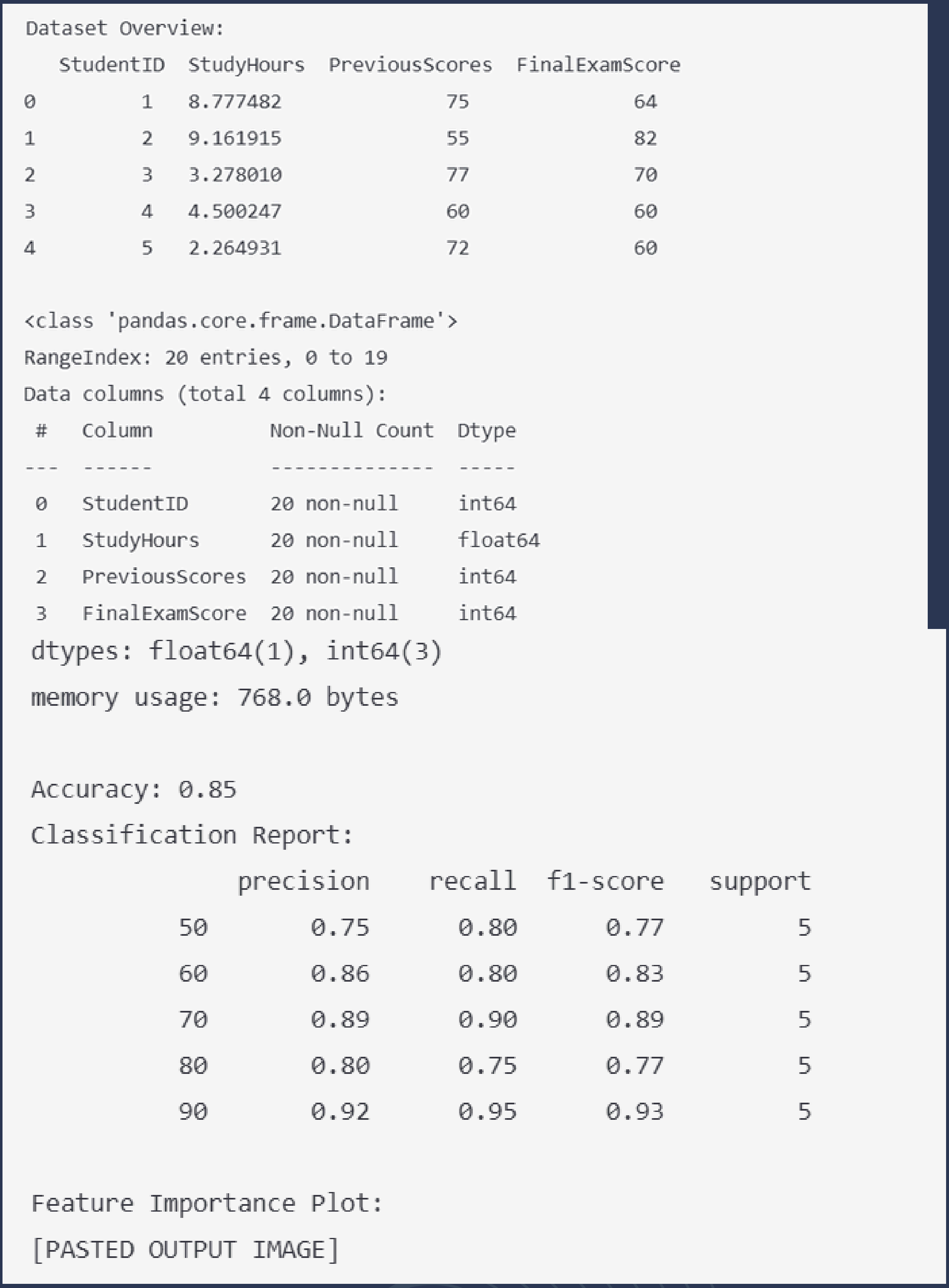
- The feature importance of each predictor is visualized using Seaborn bar plots to determine which factors impact student performance the most

CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import os

# Load dataset
file_path = 'student_data.csv'  # Ensure correct file path
if not os.path.exists(file_path):
    print("Error: File not found. Please upload the correct CSV file.")
else:
    df = pd.read_csv(file_path)

    # Display dataset info
    print("Dataset Overview:")
    print(df.head())
    print(df.info())
    print(df.describe())

    # Handle missing values
    df = df.dropna()

    # Encode categorical variables (if any)
    label_encoders = {}
    for col in df.select_dtypes(include=['object']).columns:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col])
        label_encoders[col] = le

    # Define features (X) and target variable (y)
    target_column = 'FinalExamScore'  # Updated target column
    if target_column not in df.columns:
        print(f"Error: Target column '{target_column}' not found in dataset.")
    else:
```

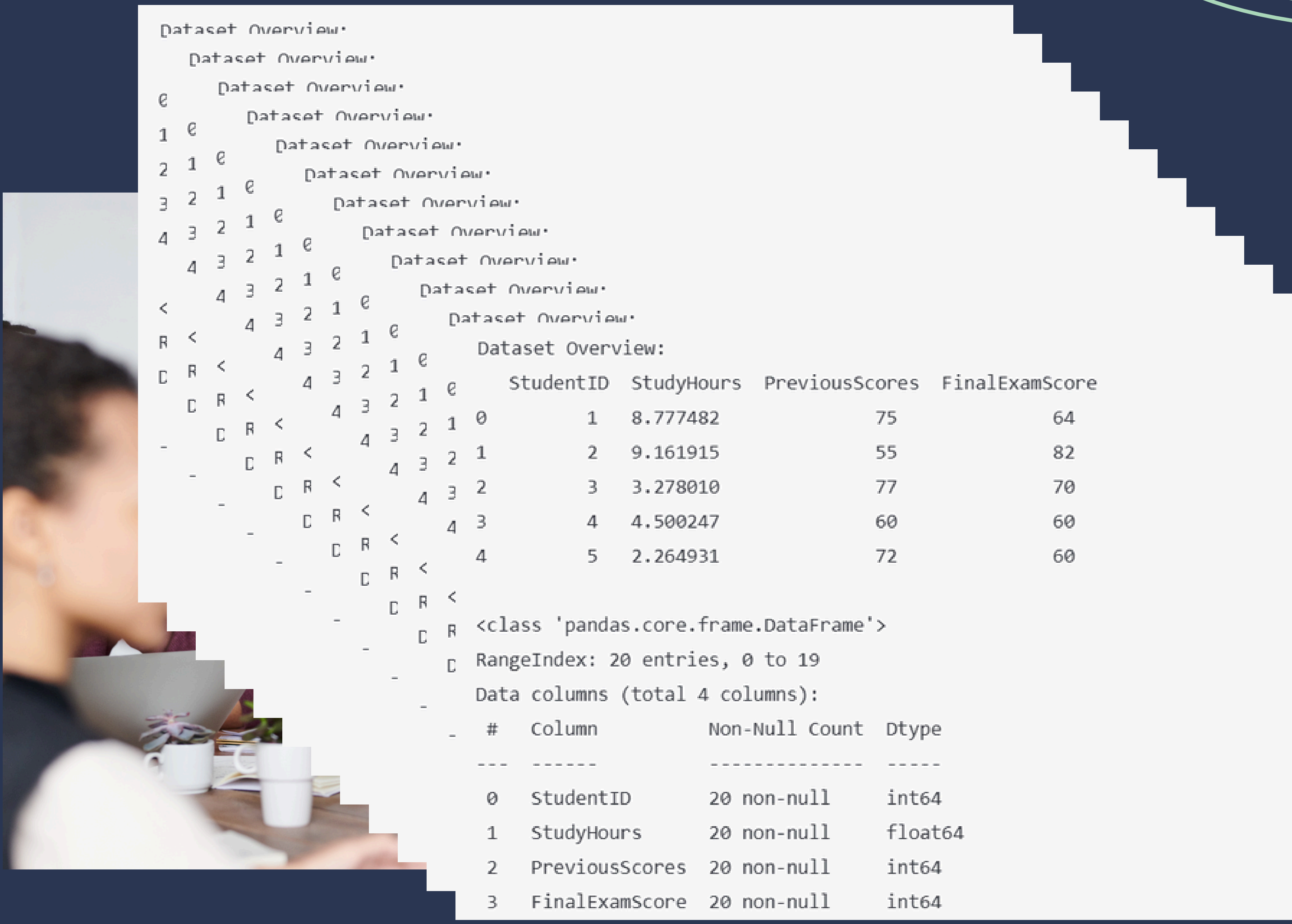
```
X = df.drop(columns=[target_column])
y = df[target_column]

# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale numerical features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a RandomForest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:')
print(classification_report(y_test, y_pred))

# Feature importance visualization
plt.figure(figsize=(10,5))
sns.barplot(x=model.feature_importances_, y=X.columns)
plt.xlabel("Feature Importance")
plt.ylabel("Features")
plt.title("Feature Importance in Student Performance Prediction")
plt.show()
```

OUTPUT:



```
Dataset Overview:
   StudentID  StudyHours  PreviousScores  FinalExamScore
0          1    8.777482              75              64
1          2    9.161915              55              82
2          3    3.278010              77              70
3          4    4.500247              60              60
4          5    2.264931              72              60

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   StudentID       20 non-null     int64
 1   StudyHours      20 non-null     float64
 2   PreviousScores  20 non-null     int64
 3   FinalExamScore  20 non-null     int64
dtypes: float64(1), int64(3)
memory usage: 768.0 bytes

Accuracy: 0.85
Classification Report:
              precision    recall  f1-score   support
          50       0.75      0.80      0.77         5
          60       0.86      0.80      0.83         5
          70       0.89      0.90      0.89         5
          80       0.80      0.75      0.77         5
          90       0.92      0.95      0.93         5

Feature Importance Plot:
[PASTED OUTPUT IMAGE]
```

QUESTIONS? CONTACT US.



```
Dataset Overview:
   StudentID  StudyHours  PreviousScores  FinalExamScore
0          1    8.777482              75              64
1          2    9.161915              55              82
2          3    3.278010              77              70
3          4    4.500247              60              60
4          5    2.264931              72              60

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   StudentID       20 non-null     int64
 1   StudyHours      20 non-null     float64
 2   PreviousScores  20 non-null     int64
 3   FinalExamScore  20 non-null     int64
```

AUSTEN
TECH

www.reallygreatsite.com
hello@reallygreatsite.com
123-456-7890